DOCUMENT RESUME

ED 321 746 IR 014 513

AUTHOR	Lewis, R., Ed.; Mace, Terry, Ed.
TITLE	Authoring Tools for Simulation-Based CBT--an Interim Project Report. Occasional Paper InTER/13/89.
INSTITUTION	Economic and Social Research Council, Lancaster (England).; Lancaster Univ. (England). Dept. of Psychology.
PUB DATE	Jul 89
NOTE	24p.; A product of the Information Technology in Education Research Programme.
AVAILABLE FROM	ESRC InTER Programme, Department of Psychology, University of Lancaster, Lancaster LA1 4YF, United Kingdom (single copy) free).
PUB TYPE	Reports - Research/Technical (143)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Authoring Aids (Programing); *Computer Assisted Instruction; *Computer Simulation; *Computer Software Development; *Courseware; Foreign Countries; Higher Education; Models; *Training
IDENTIFIERS	Great Britain

ABSTRACT

This paper describes a project which aims to create an authoring environment for simulation-based training in order to illustrate features necessary for: (1) an environment which can be used by teachers/trainers with minimum technical skills; (2) rapid prototyping by all courseware producers; (3) and tools for model building. The paper begins with a section summarizing results of an earlier survey of commercial and educational producers of computer-based training materials. The second section outlines training, authoring, delivery, and development requirements for simulation courseware, and the third section lists overall project requirements. The fourth section describes the current phase of the project, i.e., the production of a demonstration toolkit. Requirements for prototype courseware are summarized in the fifth section, including general requirements, courseware functionality, and assumptions. The sixth section presents prototype model requirements, covering general requirements, typical model components, and the nature of required models. The seventh section lists demonstration phase toolkit requirements, including general requirements, modelling tools, and authoring and learner interface tools. General, modelling, and authoring requirements not addressed by the demonstrator phase are discussed in the eighth section. The appendix includes a description of two courseware products to be produced in the demonstration phase and a list of the abstract properties of model objects. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED321746

ESRC
ECONOMIC AND SOCIAL RESEARCH COUNCIL

TRAINING AGENCY

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Authoring Tools for simulation – based CBT an interim project report

Occasional Paper: InTER/13/89
July 1989

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
R. Lewis
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Edited by:
PROFESSOR R. LEWIS
UNIVERSITY OF LANCASTER
and
TERRY MACE
INFORMATION LEARNING SYSTEMS LTD.

IR 014 513

INTER 2
INFORMATION TECHNOLOGY IN EDUCATION RESEARCH PROGRAMME
DEPARTMENT OF PSYCHOLOGY, UNIVERSITY OF LANCASTER, LA1 4YF

Origins of the ESRC INFORMATION TECHNOLOGY AND EDUCATION PROGRAMME

The Education and Human Development Committee was established with the reorganisation of the then Social Science Research Council in May 1982. In 1984 the Council changed its name to the Economic and Social Research Council. Early in 1983 the Committee identified and circulated for discussion an initial listing of important topics which warranted expanded support or accelerated development. The broad area of Information Technology in Education occupied a prominent place in that list. The Committee emphasised its intention that research would be centred not only on the effect on education of machines to help teach the existing curriculum, but on the development and adaptation of the curriculum to equip people, including those of school age, to deal with intelligent machines and to prepare them for a life changed by their arrival. For example, there are questions concerning both cognitive and organisational factors which facilitate or inhibit the adoption of Information Technology in Education, and allied to these, questions around the nature, characteristics and development of information technology literacy. These initial topics remain central to the Committee's projected agenda.

Two reports were commissioned and detailed discussion and workshops were held in 1983. In its further considerations, the Committee was conscious of the fact that the research community is widely scattered and has relatively few large groups of researchers. Furthermore, it recognised the importance of involving practitioners and policy makers in the development of its programme of substantive research and research related activities and the necessity of ensuring close collaboration with commercial organisations such as publishers, software houses and hardware manufacturers. It was this thinking that led the Committee away from the establishment of a single new centre to the appointment of a coordinator as the focal point for the development of the initiative throughout the country.

The brief for the Coordinator included:

- the review, evaluation and dissemination of the recent and current activity in the field of Information Technology and Education;
- the identification of the needs of education in relation to Information Technology;
- the stimulation of relevant research and the formulation of research guidelines;
- the establishment and maintenance of a database of relevant work and undertaking arrangements for coordinating and networking of those active in the field including cognitive scientists, educational researchers, practitioners and policymakers.

In January 1988 the Council of ESRC approved a new initiative which would have resources to support a substantive research programme. This programme, the Information Technology in Education Research Programme, gets underway in the autumn of 1988. A new series of InTER Programme Occasional Papers will begin to appear in a similar format to the current ITE Programme series. The latter are listed on the back cover of this paper.

Authoring Environments for Simulation – based CBT – a progress report

PREFACE

This Occasional Paper provides a current account of a Demonstrator Project largely financed by the Learning Technology Unit of the Training Agency. It followed a feasibility study which took the form of a survey of commercial and academic producers of computer – based learning materials. Details of this preliminary work are described in the paper.

This Demonstrator Phase of a full implementation project started in January 1989 and extends over one year. Three months into the project the first milestone, a "Project Requirements Specification" (PRS), was prepared. A large portion of this paper is based on that PRS which is currently guiding the development of the authoring and courseware demonstrators.

The current work is of particular interest to the InTER Programme as it is identifying and marking specific a large number of future research issues which can, at this time, only be dealt with pragmatically. A number of such issues were raised at an InTER seminar which is reported in the Occasional Paper InTER/7/88 and in ESRC/CNRS workshops during 1988. Taken together these approaches are raising fundamental issues, until now scarcely addressed, over the definition of authoring requirements for a new generation of tools and systems. It will be interesting to see how many such issues are manifest in the outcomes of the current round of CEC DELTA projects.

Terry Mace, the Project Manager, and I are grateful to the Project team members and to the Training Agency for agreeing to the publication of this paper. Those concerned are:

Dr Terry Hinton (University of Surrey), Dr Trevor Hopkins (University of Manchester), John Lougher (British Steel plc), Richard Millwood (King's College London), Jeff Oliver (Castle Learning Systems), Ian Graham and Janet Rothwell (Mentor Interactive Training Ltd), Jon Stock (Crosfield Electronics) and Dr. John Gillingham (Training Agency).

We are also grateful for advice on the Project being given by the external members of the Steering Group, John Haberfield (BP Research International) as chairman, and Paul Chapman (ICI Group Research).

Professor R. Lewis
ESRC – InTER Programme
University of Lancaster
June 1989

CONTENTS

1. Background page 3

2. Project Outline page 5
   - 2.1 Introduction
   - 2.2 Training requirements
   - 2.3 Authoring requirements
   - 2.4 Delivery requirements
   - 2.5 Development requirements

3. Overall Project Requirements page 6

4. Demonstrator Phase page 8

5. Summary of Requirements for Prototype Course page 8
   - 5.1 General requirements
   - 5.2 Courseware functionality
   - 5.3 Assumptions

6. Prototype Model Requirements page 10
   - 6.1 General requirements
   - 6.2 Typical model components
   - 6.3 Nature of required models

7. Demonstrator Phase Toolkit Requirements page 12
   - 7.1 General requirements
   - 7.2 Modelling tools
   - 7.3 Authoring and learner interface tools

8. Project Requirements not addressed by the Demonstrator Phase page 14
   - 8.1 General requirements
   - 8.2 Modelling requirements
   - 8.3 Authoring requirements

9. Postcript page 15

Appendix A: Courseware Requirements page 16

Appendix B: Modelling Objects page 19

1. BACKGROUND

A survey of commercial and educational producers of Computer-Based Training (CBT) materials was undertaken in March 1988. The project aims were to review existing methods of CBT provision in the training marketplace, identifying systems and practices already in place and exploring the potential for CBT simulation, and authoring tools to support it. The objective was to pave the way for the preparation of a specification for new and enhanced systems for producing CBT.

The project aimed to provide information about:

- the state-of-the-art of CBT authoring systems;
- the features which exist in current systems;
- other software packages, currently available on microcomputers which could have relevance to authoring (for example, graphics editors, ideas processors, modelling programs);
- how current authoring systems could be enhanced;
- the kinds of tools that CBT authors and trainers would like to have at their disposal;
- how closely their description of these tools dovetail with the views outlined in the project proposal.

A sample survey of current and potential producers of CBT was undertaken in order to establish the extent to which they used authoring tools, the characteristics of those tools and the perceived need for tools to support the development of simulation-based CBT. Primary methods chosen to implement the study were interviews, conducted with CBT designers and the training coordinators in major companies, combined with a video shown during the interviews which was composed of extracts of different software packages selected to illustrate aspects of simulation and authoring environments.

The twenty interviews conducted yielded information on the following:

- CBT systems (and authoring systems) in use;
- opinions regarding the appropriate use of CBT;
- in- and out-of-house provision of CBT;
- features of the commercial authoring systems in use;
- details regarding the CBT design and implementation process;
- current uses of simulation in training;
- opinions regarding the use of simulation in training.

Informed by the results of the interviews, the following issues should be considered in any specification of an authoring system for CBT which allows for the development of simulation-based elements:

- the effective use of a simulation requires that the trainee have sufficient prior knowledge to be able to make reasoned decisions;
- effective simulation within CBT should incorporate some form of tutorial dialogue which allows the trainee to interact with the system;
- the software should have the ability to monitor the trainee's progress;
- simulations to date have been purely algorithmic; what is required for the future is that simulations be capable of reasoning over the knowledge domain which the simulation represents.

Current authoring systems are found to be adequate for the job they were intended to do – that is, providing tutorial material and animations. We have not, however, been able to find any low-cost, widely-available authoring systems which are adequate in allowing courseware developers to incorporate simulations based on a mathematical or 'behavioural' model in a cost-effective way. Even the high

cost alternatives fall short in that they are difficult to use and do not contain the necessary features. In particular, the trainee needs to be able to use the resultant software in ways which the designer could not anticipate and see the effects of those actions, as well as follow the prescribed actions.

The biggest problem faced by professional courseware developers, however, and the area in which tools are most needed, is in the design area. This covers the initial knowledge structuring, the task analysis and course logic layout. This need arises particularly when a complete course is being planned.

There is also a large problem in the area of 'prototyping' – of being able to generate mock-ups of the final screens so that the developer and the user can quickly make the inevitable changes without having to go through a lengthy specification stage on paper. Once they become engaged in designing simulations, developers will also find that they need model building tools. If software support were available in both these areas, communication between members of a development team (and between the team and their client, when that situation exists) would be greatly improved and productivity would rise rapidly.

Any development of authoring tools should be for generally available hardware and software environments. At this time, the IBM PC/MS-DOS environment is the most common; it is important to be alert to future change.

The full report was printed as ITE/27/88 from the ESRC – ITE Programme and later as "Authoring of Computer Based Training Materials" from the Training Agency. Following this, an Outline Functional Specification for the required system was produced as an internal document. An ESRC – InTER Programme seminar on "Support Tools for Authoring" was reported in InTER/7/89.

Some working definitions

In this document, a *simulation* consists of a *model* – – the computer representation of the system, process or machine being simulated – – and a *learner interface*, which is capable of presenting aspects of the model to the learner and permits the status of the model to be affected by the learner. A model contains inter – related *components*, which may be in one of a variety of different states defined by the *values* of its *attributes* or *parameters*.

The *courseware* reflects the training requirements and strategy. It will necessarily incorporate a simulation, and may include much other material, such as help or explanatory text and graphics.

There are several different *roles* referred to in this paper. They may be separate individuals or multiple roles may be undertaken by one person.

The terms *trainee* and *learner* are used interchangeably to indicate the final recipient of a computer – based training (CBT) course, except that the recipients of the prototype courseware happen to be *trainees*. The *trainer* is the person who identifies the training need, and is responsible for satisfying that need.

The person constructing the model is called the *modeller*. Specialist knowledge required during model building will come from a *subject matter expert*. The person constructing the learner interface to the model is called the *learner interface builder*.

The person constructing a CBT course based on a simulation is called the *courseware author*, or just the *author*. The term *developer* covers modellers, learner interface builders, and authors.

2. PROJECT OUTLINE

2.1 Introduction

Although, as described in the previous reports, simulations may be used in training in many ways, one area where they have proved particularly effective is in the teaching of fault - diagnosis. This technique may be applied, for example, to an individual machine, (which has components which may fail or perform outside specification), or to a manufacturing production process, where the desired outcome is achieved only when everything is adjusted within prescribed limits.

A computer - based simulation of a system has several advantages over alternative methods of training in such cases, for example, a simplified view can be presented, enabling the trainee to concentrate on the most important aspects of the machine or process, without being distracted by irrelevant details. Fundamentally, however, simulations offer the opportunity to *learn by doing*: to reinforce theoretical knowledge through problem solving in a context similar to that faced in reality. In most cases, they can generate a wider variety of fault conditions than can be produced cost - effectively on the real equipment.

However, in order to allow the development of fault - finding skills, the system must faithfully represent the major aspects of the appearance and behaviour of the system under consideration, and react in a realistic manner when the trainee changes the current state of the model. It must be able to offer information about the components and be able to guide the trainee, where necessary, to the solution of the problem.

2.2 Training requirements

In order to be effective, courseware incorporating simulations must:

- provide the trainee with tutorial support explaining the theoretical basis of the machine or production process;
- give the trainee instructions and explanations of the particular model which forms the basis of the training session;
- represent the actual appearance of the real system, (simplified as appropriate and within the limitations of screen display);
- allow the trainee to examine the current status of the components of the machine or production process;
- allow the trainee, by interacting with the model, to affect its status in a realistic way;
- provide help and guidance to the trainee on request;
- be capable of providing the trainer with feedback on trainee achievement.

2.3 Authoring requirements

The creation of courseware to meet the above training requirements demands an authoring environment (or toolkit) with certain features.

- In order to represent, with reasonable fidelity, both the appearance and behaviour of the system under consideration, the tools provided to the courseware producer must allow the definition of sophisticated graphics and the ability to define complex interactions between component parts of the machine or process.

- The toolkit must provide a general environment which will allow the definition of models for a wide range of differing situations. To this end, the facilities provided for defining the behaviour of the components must cover steady-state, time-based and event-based behaviour, and individual models may contain elements of all three.
- There must be a clear distinction between the "model" (the underlying behaviour, defined in mathematical or other terms) and the "view" of the subject presented to the trainee.
- The system must support the provision of tutorial material in a "layered" or "Hypertext" format, so that the trainee can gain extra information on words or phrases on the screen by selecting them (using the mouse, for example).
- The toolkit must be usable by courseware producers who will not necessarily have previous experience of building simulations, or of using them as part of a training programme.
- The toolkit must be adaptable for use by both novice and experienced courseware producers.
- The toolkit must be designed, as far as possible, in such a way as to facilitate links with existing utilities.

2.4 Delivery requirements

- The delivery of courseware will be on hardware commonly available to learners, currently characterised by the PC.
- The authoring environment will require greater processing capability, as characterised currently by 386-based PCs. The output courseware will be suitable for direct implementation on the learner delivery hardware.

2.5 Development requirements

- Powerful workstations are needed to design and implement the authoring environment which must be capable of direct implementation on the target authoring hardware. SUN workstations are being used.

3. OVERALL PROJECT REQUIREMENTS

These are summarised as follows:

- The project is required to develop, evaluate, and subsequently produce a releasable version of a set of advanced software tools, referred to as the toolkit. These tools are to enable the more effective provision of courseware featuring the extensive use of simulations.
- The courseware produced by the use of the toolkit is intended to reinforce theoretical knowledge through problem-solving in a context similar to that faced in reality.
- The toolkit must substantially reduce the time (and thereby the cost) of the development of such courseware compared with existing authoring systems and languages.

- In particular, the toolkit must allow developers the ability to construct software models of physical systems or processes, and to present images generated by such models as the basis of fault diagnosis exercises.
- The model must be capable of being used in different ways by the author, depending on the defined training strategy.
- The courseware must be able to present tutorial material to explain the task which the trainee is required to perform. This task will most frequently be to return the model to a fault - free condition by identifying, replacing or adjusting (where appropriate) defective or incorrectly adjusted components.
- The courseware must be able to represent the actual appearance of the real equipment or process (simplified as appropriate, and within the limitations of the available computer hardware).
- The courseware must allow the trainee, by interacting with the learner interface, to affect the model's status in a realistic way.
- The courseware must be able to offer information about the state of the components of the model.
- The courseware must be able to guide the trainee, where necessary, in the performance of the task.
- The toolkit must be capable of providing the trainer with data related to trainee actions.
- The toolkit must allow the definition of sophisticated graphics and the ability to define complex interactions between components.
- The toolkit must provide a general environment which will allow the definition of models for a wide range of differing situations. To this end, the facilities provided for defining the behaviour of the components must cover steady - state, time - based, and event - based behaviour. Individual models may contain elements of all three.
- The courseware must support the provision of online help in a hypertext format, so that the trainee can gain extra information by selecting items.
- The toolkit must be usable by both novice and experienced developers, including those who do not have previous experience of building simulations, or of using them as part of a training programme.
- The toolkit must be designed in such a way as to facilitate links with other utilities and languages.
- The delivery of courseware will be on hardware commonly available to learners, currently characterised by the PC.
- The toolkit will require greater processing capability, as characterised currently by 386 - based PCs.

4. DEMONSTRATOR PHASE

The initial (current) 13-month phase of the project is required to produce a demonstrator toolkit. This will be used to help establish the detailed functionality of the final toolkit, demonstrate applicable techniques, suggested screen layouts and styles of interaction, and assess the efficacy of the training approach.

The demonstrator toolkit is required to undergo trials by courseware authors, and by trainees and trainers in two customer sites. The resulting feedback will form the basis for the detailed specification of the final toolkit. The two training tasks, defined by Ring Rolled Products Ltd., a subsidiary company of British Steel, and by Crosfield Electronics, are outlined in Appendix A.

The prototype toolkit is required to demonstrate the key aspects of the functionality of the final toolkit, on the development hardware. During the later stages of the demonstrator phase, a substantial part of the toolkit will be implemented on a 386-based PC.

In the following sections various requirements for this phase of the project are outlined. The first (Section 5) describes the end-user (learner) needs; this is followed by an analysis (Section 6) of the requirements of the models of the systems being simulated. Finally (Section 7) draw on the consequences of the two preceding sections in defining the necessary authoring tools

5. SUMMARY OF REQUIREMENTS FOR PROTOTYPE COURSEWARE

5.1 General requirements

- The courseware is required to allow the trainees additional practice in fault-finding skills.
- The courseware will allow the trainees to interact with a simulation, based upon a model of the machine under consideration.
- The trainee will require to perform repeated operational cycles of the model, and view and compare the outputs from each cycle.
- The trainer will require access to data relating to the actions taken by the trainee whilst using the courseware.
- The courseware is required to indicate to the trainee the time implications of any action taken, if required by the author. The total diagnosis time may be indicated whilst the system is running.
- The prototype courseware is required to be delivered on Sun workstations, for User Trial in October, 1989.
- By the end of the demonstrator phase, a substantial portion of the toolkit and prototype courseware will be demonstrable on a 386-based PC.

5.2 *Courseware functionality*

All trainee functionality will be implemented by the author, depending on the particular training strategies. The trainee may require:

- access to a diagrammatic representation of the components of the model and their inter - relationships;
- to measure or monitor parameters of some components in the model;
- to replace suspected faulty components in the model;
- access to context - sensitive help whilst using the courseware;
- to undo the effects of previous decisions, so as to return to one of a limited number of checkpoints.

The trainee may also be able to adjust the settings of components of the simulation, within pre - defined limits. When the trainee replaces a component, the replacement component is assumed to be fault - free.

There are a number of requirements specific to one or other of the pieces of courseware, for example:

- Each symptom presented to the trainee may be caused by a combination of faults. In particular, simultaneous hydraulic and electrical faults may be simulated. (Ring Rolled Products)
- The courseware is not required to suggest, or include any reference to, a best method of solving a fault condition. (Crosfield)
- Each symptom presented to the trainee will be caused by one fault only. (Crosfield)

5.3 *Assumptions*

- The trainee will already have received plant - specific training and instruction in fault - finding strategies.
- Project members will have access to details of the machine component behaviour.
- Crosfield and British Steel personnel can provide project members with details of the symptoms required to be reproduced, and a number of faults which would produce each symptom.
- The symptoms can be reproduced with reasonable fidelity using the hardware available to the project (see Appendix A).
- Someone familiar with the courseware will be present when it is being used.
- A functional version of the toolkit, running on the development hardware, will be available at least two months before the agreed date of the commencement of the customer trials.

6. PROTOTYPE MODEL REQUIREMENTS

6.1 General requirements

Both models will require general-purpose *modelling objects*. These objects will be named, and must be capable of supporting appropriate behaviour. Connections between these objects must be definable (see Appendix B). Both models will be constructed from a collection of these objects. It must be possible to specify both a 'fault-free' and, if required, one or more 'faulty' behaviours for any object.

The models for both the Magnascan scanner and the ring blank press are of the *single-shot, discrete-event* type. In this kind of model, when parameters are changed (e.g.a printed circuit board is replaced, or a different machine parameter is used for a run), then these changes are propagated once throughout the model, and affect the outputs.

It does not appear to be necessary to model time explicitly in either case. Events in the models are partially-ordered, but do not have a particular time interval associated with them.

Both models are required to support an 'undo' facility, allowing the state of the model to be returned to a previous condition. Furthermore, it must be possible for the model to be put into one of a range of pre-determined conditions.

6.2 Typical model components

- hydraulic components (such as reservoirs, logic elements, pipes and non-return valves);
- electro-mechanical components (such as limit switches, leadscrew motors);
- analogue electrical components (such as lamps, switches, wires, relays and contact blocks);
- digital electronics (such as the PLC, I/O boards, signal processors);

6.3 Nature of required models

The Ring Blank Press model will include a representation of the PLC, which will provide overall control in a similar way to the real component. It appears to be possible to model in adequate detail the entire press. Although this involves a large number of separate components (perhaps around 100), most of these are of the types listed above. Many of these types of component are used in a wide range of British Steel plant.

In general, the control of the Press is viewed in terms of interacting mechanical, hydraulic and electrical parts. However, the PLC embodies this control in terms of a program. It appears that the best way of modelling this would be to construct a single modelling object corresponding to the PLC, with many inputs and outputs, and encode a representation of the PLC program in a modelling language.

Most of the information passed between the modelling objects will signify only that particular events have occurred. However, there may be some cases where either digitized or analogue information is communicated.

In general terms, the Magnascan scanner can be regarded as performing a large number of processing stages on information captured from the original image. In some cases, this information is represented by a stream of 8-bit digital values,

while in other cases, it consists of varying physical quantities (such as voltage or light intensity). This suggests that the model should consist of a collection of objects representing these processing stages, communicating events which may have associated values. In general, a printed circuit board within the scanner corresponds to a small number (often one) of these objects.

Thus, the proposed model consists of a small number (perhaps 25) of interacting modelling objects. However, most of these objects will have relatively complex behaviour, so that a major part of the modeller's task will be in representing (in a modelling language) the behaviour of each of these objects. Each object will be, in general, different and heavily specialised to the requirements of this model.

During the Demonstrator Phase, it will not be possible to construct a simulation of the entire scanner, in the time available. Since it is necessary to model the expose machine helical scan mechanism in some detail in order to synthesise film faults, only the expose machine will be modelled in detail. This implies that, while a large number of different faults in the expose machine can be modelled, no faults in the analyse machine can be represented.

7. DEMONSTRATOR PHASE TOOLKIT REQUIREMENTS

This section first deals with the requirements for tools to build the prototype models, and then with tools for constructing the learner interface as well as the courseware.

7.1 General requirements

- The tools to support both simulation construction and authoring must employ a consistent and integrated metaphor, as far as is possible. A developer must be able to move between modelling, interface building and authoring with the minimum of re–orientation.
- It must be possible for the developer to move easily to and from an emulation of the learner's environment.
- The functionality of the Phase I toolkit will be based on existing *SMALLTALK* tools and applications wherever possible. To this extent, independent, potentially overlapping windows, iconic representations, a pointing device (probably using a mouse) and pop–up or pull–down menus will be used.
- The developer must be able to configure his or her own method of interaction with the toolkit, as well as the learner's interface with the courseware.

7.2 Modelling Tools

In general, the requirements for constructing models arise from two sets of requirements:

- from the requirement to represent the actual equipment adequately;
- from the identified training requirements.

The toolkit must support the iterative development of the model, allowing both sets of requirements to be managed simultaneously.

After design and implementation of the model, it will be necessary to verify it against both sets of requirements.

The model construction tools must provide simpler, easier to use yet more powerful facilities than *SMALLTALK*, and provide the modeller with specialised tools. These facilities must include:

- A tool for creating new kinds of *primitive object* – those whose behaviour is defined using only the *modelling language*.
- Wherever possible in the available time, higher–level tools will be provided to define the behaviour, which will in turn generate the necessary 'program code'. The resultant code will remain accessible to the developers.
- A tool to specialise or customise the behaviour of kinds of primitive objects. It should be possible to 'browse' a *library* of existing kinds of objects, and selectively create new objects.
- A tool for constructing composite objects from other objects (sub–objects), with a direct–manipulation interface allowing the modeller to describe graphically the connections between objects. This tool will allow relatively inexperienced modellers to construct a model using a library of existing kinds of objects.

- Tools allowing the behaviour of existing objects to be modified and debugged interactively.
- A tool which allows a model to be verified independently of any courseware which might use that model.
- A tool to support documentation of the model. This documentation is to be available both on - and off - line.

[See Appendix B for a conceptual basis for modelling]

7.3 Authoring and Learner Interface Tools

The following tools are required by the author and learner interface builder:

- Tools to edit text and colour graphics using a direct - manipulation interface. This will include facilities to import from, and export to, other utilities.
- Tools to define the connections between objects in the model and parts of the learner interface, which will determine the appearance of the resultant display;
- Tools to allow the definition of the trainee's access to the components of the model (and, where appropriate, their parameters), including simulated input and output devices;
- Tools to define the actions or tests that will be available to the trainee;
- Tools to allow the specification of help material;
- Tools which can be used to express the relationships between courseware entities.
- Tools to define, at any point, the state of the components of the model;
- Tools to control the accumulation and display of parameters of the model, such as the diagnostic time;
- Tools to specify which 'variables' can be monitored by the trainee;
- Tools to support the use of checkpoints;
- Tools to view an emulation of the learner environment;
- Tools which prompt for trainee input, and validate it;
- Tools which support the creation of documentation.

8. PROJECT REQUIREMENTS NOT ADDRESSED BY THE DEMONSTRATOR PHASE

8.1 General requirements

This Phase will help establish a methodology for the use of the toolkit. For example, the process of defining the model in terms of the training and modelling needs may be an iterative process. There may be some opportunity to produce tools in the subsequent phase which will support and help to structure this methodology.

8.2 Modelling requirements

- The toolkit must be capable of supporting the construction of simulations involving steady-state and rule-based behaviour. The toolkit must also be capable of constructing 'mixed-mode' models. Detailed requirements for this must be formulated during the Demonstrator Phase.
- The provision of higher-level tools to define the behaviour of modelling objects will be addressed more fully later. In the demonstrator phase, the emphasis will be on making the functionality available.
- The prototype toolkit will not support the re-design of the model by the trainee. This may be required in certain applications.
- The model, once defined, must be able to be maintained easily. Thus, where the behaviour is defined by referencing program code, these must be accessible through a higher level interface. The system should, as far as is possible, be self-documenting.
- Interfaces to a large number of other utilities and programming languages will be supported by the final toolkit. These will be defined when a decision is taken on the implementation strategy for the final toolkit.

8.3 Authoring requirements

- Although the provision of context-sensitive help is included in the Demonstrator Phase, the requirement for a worked solution will not be addressed. In the later phase, some level of solution, generated wholly- or semi-automatically, will be implemented.
- The models developed for the two sets of prototype courseware are only required to be used as the basis for fault-diagnosis exercises. It must also be possible to use the simulations in demonstration mode to explain the operation of the machine or process, and to allow the learner to explore the underlying behaviour.
- The resulting simulations may run too quickly, or too slowly, to support the training need. The developer may require to control this aspect of the courseware.
- Only symptoms due to faulty components are required to be simulated in the prototype courseware. More generally, the trainee may require to alter settings or parameters of components, often by selecting from a list of pre-defined 'states'.
- There may be occasions when the simulation is forced to halt because parameter(s) exceed pre-defined ranges. In this case, the author will require facilities to specify the display of help or explanation material, or take other actions.

9. POSTCRIPT

As might be expected, the most challenging aspects of The Demonstrator Phase lie in reconciling the needs of training, which sometimes become too ambitious, with the practicalities of the software engineering. The courseware design team sit in the middle as arbitrators!

To date, the Project is on schedule and it is expected that, despite their ambitious nature, the targets set will be achieved. The team is aware that in some respects it is breaking new ground, whilst in danger of reinventing yet another authoring system.

Current design discussions relate to the key functions of producing tools which assist the author in creating, testing and modifying the model of the system. The second critical area, in many respects more complex, is that of supporting the author in creating a flexible (learner adaptable) interface, between learner and model. In both cases it is essential that coherent explicit yet adaptable, structures are implemented. The lessons learnt, in particular the current analysis of model building, will be documented in future papers.

As more and more powerful software becomes available for more and more easily available hardware, a barrier to the use of that software is often to be found in the human user, in this case authors. This is not to say that human users are unimaginative, it is simply the problem of linking their existing experience with ways of making this explicit. Part of the challenge of the Project is to provide software which bridges that gap; simple metaphors for the software functionality are required which make available powerful tools to address real training needs.

APPENDIX A: COURSEWARE REQUIREMENTS

In order to validate the functionality of the toolkit produced in the Demonstrator Phase of the project, two pieces of courseware will be produced, for *Ring Rolled Products* and *Crosfield Electronics*. The background to each of the requirements is given below, and is followed by a consideration of generic and customer-specific requirements arising from them.

a) Ring Rolled Products

Background

Ring Rolled Products Ltd.is a wholly-owned subsidiary company of British Steel plc. It produces rolled steel rings for applications such as pipe flanges, roller bearing cases and tyres for rolling stock.

British Steel has a progressive approach to training and has its own CBT/IV development and production team within the Open Learning Development Unit (OLDU). Many companies within British Steel also have their own training centres. The OLDU is part of the Central Training Unit, a reservoir of training expertise which can be utilised by other training centres.

Ring Rolled Products is a relatively small plant employing about two hundred workers in three shifts. It does not have its own training centre. Line managers undertake on-the-job training when necessary.

Overview of Press

The Ring Blank Press is a three-stage 1600 tonne hydraulic press which forms a hot steel billet into a ring blank for later processing in the rolling mill. The three stages are 'upsetting', 'pressing' and 'punching'.

The motions of the major parts of the press are powered by hydraulic rams, with motive power from six pumps driven by three 250 kilowatt electric motors. There are a large number of electrically-operated (solenoid) valves, controlled by a sophisticated Programmable Logic Controller (PLC). Feedback from the moving parts of the press to the PLC is by limit switches and other sensors, including pressure switches and digital position sensors.

The Current Problem

Due to a number of factors including its inherent complexity and the hostile environment in which it operates, the press is prone to frequent breakdown. Each time the press fails to operate correctly, a fitter and/or electrician is called, whose task is to locate the cause of the problem and rectify the fault.

A major production cost is in the 'down-time' caused by failures; any measure which reduces down-time is likely to be cost-effective. The difficulty of fault diagnosis is in some part due to the interfacing of different systems and in deciding which part is producing the malfunction. However, the craftsmen responsible for the first-line maintenance of the press generally have skills in one particular area, and are not used to dealing with the system as an integrated whole. This means that fault diagnosis can sometimes be protracted.

Trainee Profile

Typically, the trainee will be male and between the ages of 30 and 50.

He will be a skilled craftsman who served his four-year apprenticeship with British Steel. He will belong to a trade union.

All craftsman will have received plant-specific training. In addition, fitters will have had training in hydraulics and basic multi-skill training; electricians will have undergone training in PLC's, basic electronics and computer awareness. In general, the electricians are more highly trained than the fitters, but both the fitters and electricians have academic qualifications to BTEC standard. The maintenance team consists of approximately 12 people.

The Training Need
The training need is to provide practice in the techniques of diagnosing which component or components of the press are faulty, given a particular symptom, in order to encourage the trainee to view the press as a dynamic system of inter-related mechanical, hydraulic and electrical components.

Training Objective
At the end of the press training (including tutor-led elements), the trainee will be able to diagnose pre-defined press faults irrespective of their own craft discipline. A time saving of 50 is the target to be achieved for the diagnosis of cross-discipline faults and 25 for faults in their own discipline.

The standard of performance will be measured by the time that the trainees chosen diagnostic actions would take compared with their current performance as reflected in the recorded down-time.

b) Crosfield Electronics

Background
Crosfield Electronics is a manufacturer of high technology pre-press printing equipment. It is an international company with particular interests in the United States and Europe. Customer Service is regarded as an integral part of the Company ethic with education and training being dealt with in a highly structured and prescribed manner.

The Scanner is concerned with the production of the four monochrome transparent acetate films necessary for colour printing, from which printing plates are obtained. It is a highly sophisticated interaction of electronic, mechanical and optical components.

Scanner product training is given to Field Service engineers, Operator installers and Customers; the proposed training is targeted at the engineers. All engineers attend a formal 32-day course; this is run at Crosfield's Education and Training Centre in Watford. Each course has a maximum of twelve trainees, with four scanners available during the course. Typically, this course runs six or seven times a year.

Overview of Magnascan Scanner
The purpose of the 'Magnascan' range of machines is to make four-colour separations of photographs and transparencies, as part of the pre-press process in colour reproduction. In essence, the original is helically scanned (in the analyse machine), the colours are separated electronically, and each of the four colours is printed onto a different area of photographic film, using synchronised scanning on the expose machine. The expose process uses modulated laser light and complex optics to draw variable area dots corresponding to the density of that particular colour on the original image.

A wide range of image processing tasks can be carried out between the analyse and expose stages. The image can be enlarged or reduced in size, by varying the

relative scan rates on the two machines. The colour balance can be changed. Un-Sharp Masking (USM) can be used to emphasise edges in the original image. Dots of different shape (circular, square and oval) can be created; typically, dots representing different colours (cyan, magenta, yellow and black) are of the same shape, but different orientations.

The Current Problem

A small part of the present training programme is concerned with diagnostic problem-solving from film faults. It attempts to prepare the engineer for the more obvious scanner faults likely to be encountered in the field.

It is recognised by Crosfields that insufficient time is available for engineers to practice diagnostic fault-finding on the training course as it is presently structured. This does not suggest that there is a deficiency in the formal teaching method nor that they need training in fault-finding strategies. The synthesis of film faults is seen as a reinforcement of the knowledge gained on the present course.

Trainee Profile

Engineers come from all countries and are likely to regard English as a foreign language. All training is conducted in English and each trainee is expected to show some understanding of the language.

There is no 'typical' trainee profile but they share some common attributes. Each is able to demonstrate a competence in electronics, or a related discipline, up to HNC level or its equivalent. No knowledge of mechanics or optics is assumed at the commencement of the course. It is assumed, however, that the trainee will have received training in fault-finding strategy before using the simulation. In addition, it is assumed that the trainee will be familiar with a block diagram level description of the machine and the functionality of the major components.

The Training Need

Fault diagnosis is a necessary skill for the engineer. There is a clear opportunity to enhance the current training by providing a mechanism for these skills to be practiced. Crosfield engineers are currently trained in film fault diagnosis as part of their existing course, but this skill is soon lost if it is not frequently used, and only recovers as the engineer gains experience. Time, and the difficulty of gaining access to scanners, prevents all of the faults (and combinations of faults) being covered on the course.

Training Objective

The aim is to allow the engineers to practice their fault-finding skills on a wider range of faults than available from the current training course.

APPENDIX B: MODELLING OBJECTS

One way of envisioning the desired properties of modelling tools is to consider the 'abstract' properties of model objects. A generic model component might have the following properties:

The *Type* of the model object. For example, this might be 'Hydraulic One - way Valve' or 'Beam Computer Board'.

The *Name* of this particular object. This might be 'Limit Switch E68' or 'Modulator Amplifier 3'. It will be necessary to be able to have many different, identified components of the same type in a model.

A *Description* of this object. In general, there is a need to have various documentation items attached in some way to a model object. This might include:

- a long description of this type of object;
- a short description of this particular object;
- a long description of this particular object, and its place in the model.

Sub - objects; a collection of other objects which make up, and describe the behaviour of this object.

Inputs from other objects. Some sort of connection to objects from which input information of various sorts is expected.

Outputs to other objects. A connection to objects to which outputs (results) are passed.

For bi - directional communication, a single object can be simultaneously an *input* and an *output*.

Internal *Parameters*; these represent the state of an object, and may change from moment to moment during the course of a simulation run.

Internal *Attributes*; the also represent the state of an object, but do not change during a simulation run.

It may be sensible to combine these two internal aspects; in any case, they differ only by usage.

A *Display Hook*, which is part of the learner interface. When a particular model object changes, the display hook is used to cause the graphical representation on the screen to change.

A *Function*, describing the behaviour of this object in terms of a *modelling language*. A primitive object will have a function, but no sub - objects, while a composite object will have sub - objects, but (possibly) no function.

Given this description of the underlying modelling objects, the need to set and modify the various properties outlined above can readily be seen. The remaining problem is in devising a user interface for the modeller, to allow him/her to construct sophisticated models with the minimum of effort.

It should be noted that the structure proposed above could be used to support both 'high - level design' tools, where decomposition into sub - objects, the documentation and connections are important, and 'low - level' tools, where the detailed description (either more sub - objects, or primitive functions only) is important. It may be desirable to have a different user interface for these two tasks (high - level design, low - level implementation), but the same underlying structure could be used. Alternatively, the same tool could be used.

ESRC PUBLICATIONS RELATING TO INFORMATION TECHNOLOGY AND EDUCATION

Available from:

Economic and Social Research Council
160, Great Portland Street
London W1N 6BA
Telephone: 01 – 637 1499

Microcomputers in Education – a framework for research
(Morley Sage and David J. Smith)
Published by SSRC 1983 ISBN 0 86226 125 2 Price £2

Intelligent knowledge – based Systems
– UK Social Science Research Inputs
(Linda A. Murray and John T.E. Richardson)
Published by ESRC 1984 ISBN 0 86226 108 2 Price £2

Information Technology and Education
– signposts and research directions
(Editor David J. Smith)
Published by ESRC 1984 ISBN 0 86226 152 X Price £5

From:

School Government Publishing Company
Darby House
Bletchingley Road
Merstham
Redhill, Surrey, RH1 3DN

Information and Communication Technologies:
Social Science Research and Training
Volume 1: An Overview of Research
William H Melody and Robin E Mansell
ISBN 0 866226 179 1

Volume 2: National Directory
Robin E Mansell (editor) assisted by Barbara Richards
ISBN 0 866226 184 8

Price £10 (two volumes)

From Blackwell Scientific Publications:

Trends in Computer Assisted Education – the proceedings of the 6th Conference on Computers in Higher Education held in April 1986 and supported by CET and the ITE Programme. Published January 1987.

Computer Assisted Learning in the Social Sciences and Humanities – the proceedings of the ESRC International Seminar held in April 1986. Due for publication June 1987.

[For ITE Programme papers, see outside back cover.]

Papers currently available at no cost from:

ESRC – InTER Programme, Department of Psychology,
University of Lancaster, Lancaster LA1 4YF
Telephone: 0524 – 65201 Ext 3601 (24hr. answering)
Fax: 0524 – 63806
DIALCOM: 10001:yma027
JANET: inter@uk.ac.lancs.cent1
EARN: inter%uk.ac.lancs.cent1@ac.uk
DARPA: inter%cent1.lancs.ac.uk.@cunyvm.cuny.edu
EuroKom: ESRCITE_Prog_Lancaster%eurokom.ucd.irl@euroies.uucp

ITE Series of Occasional Papers:

A full list of papers in this series is available from the Programme. Please note that some listed may be available in photocopy form only.

InTER Series of Occasional Papers:

InTER/ 1/88 The Information Technology in Education Research Programme, September 1988

InTER/ 2/88 Artificial Intelligence Applications to Learning and Training – a seminar report to the Training Commission, August 1988

InTER/ 3/88 The InTER Programme: Groupwork with Computers, October 1988

InTER/ 4/88 The InTER Programme: Conceptual Change in Science, October 1988

InTER/ 5/88 The InTER Programme: Tools for Exploratory Learning, October 1988

InTER/ 6/88 The InTER Programme: The Independent Policy Evaluation, October 1988

InTER/ 7/88 Support Tools for Authoring – a seminar report, December 1988

InTER/ 8/89 Research in Progress – update March 1989

InTER/9a/89 A Guide to the Electronic Services, May 1989

InTER/10/89 Information Technology and Language Development – a seminar report, July 1989

InTER/11/89 Barriers to Innovation – a seminar report, July 1989

InTER/12/89 Information Technology – based Open Learning – a Study report, July 1989

InTER/13/89 Authoring Tools for simulation – based CBT – an interim project report, July 1989